WE CLAIM:

1. A method for generating a new initial state that is used in producing a downlink scrambling code, comprising:

selecting a first initial state and a second initial state;

providing the first initial state to an auxiliary linear feedback shift register;

determining whether the downlink scrambling code is a secondary downlink scrambling code;

advancing the first initial state according to a secondary index when the downlink scrambling code is a secondary downlink scrambling code; and

advancing the first initial state according to a primary index such that the new initial state is produced.

2. The method of claim 1, further comprising providing the new initial state and second initial state to a first main linear feedback shift register and a second main linear feedback shift register respectively, such that the downlink scrambling code is produced by combining a set of outputs corresponding to the outputs of the first main linear feedback shift register and the second main linear feedback shift register.

3. The method of claim 1, wherein the first initial state and the second initial state are selected such that they correspond to a predetermined slot jump.

4. The method of claim 1, wherein advancing the first initial state according to a secondary index further comprises pre-running the auxiliary linear feedback shift register for a predetermined number of samples that corresponds to secondary index.

5. The method of claim 4, wherein pre-running the auxiliary linear feedback shift register is triggered according to a timing control such that the first initial state is advanced according to the secondary index and the primary index prior to operation of the first main linear feedback shift register and a second main linear feedback shift register.

6. The method of claim 1, wherein advancing the first initial state according to a primary index further comprises combining the first initial state with a calculated primary-mask that corresponds to a selected primary downlink scrambling code.

7. The method of claim 1, further comprising fixing a set of masks used in producing the downlink scrambling code, whereby memory usage is reduced.

8. A method for generating a downlink scrambling code, comprising:

selecting a first initial state and a second initial state;

providing the first initial state to an auxiliary linear feedback shift register;

producing a new initial state by advancing the first initial state according to at least one of a primary index and a secondary index; and

providing the new initial state and the second initial state to a first main linear feedback shift register and a second main linear feedback shift register respectively, such that the downlink scrambling code is produced by combining a set of outputs corresponding to outputs of the first main linear feedback shift register and the second main linear feedback shift register.

9. The method of claim 8, wherein the first initial state and the second initial state are selected such that they correspond to a predetermined slot jump.

10. The method of claim 8, wherein advancing the first initial state according to a secondary index further comprises pre-running the auxiliary linear feedback shift register for a predetermined number of samples that corresponds to secondary index.

11. The method of claim 10, wherein pre-running the auxiliary linear feedback shift register is triggered according to a timing control such that the first initial state is advanced according to the secondary index and the primary index prior to operation of the first main linear feedback shift register and a second main linear feedback shift register.

12. The method of claim 8, wherein advancing the first initial state according to a primary index further comprises combining the first initial state with a calculated primary-mask that corresponds to a selected primary downlink scrambling code.

13. The method of claim 8, wherein the outputs of the first main linear feedback shift register and the second main linear feedback shift register are provided to at least one of a set of fixed masks to produce the set of outputs that are combined to produce the downlink scrambling code.

14. A downlink scrambling code generator, comprising:

an auxiliary linear feedback shift register that is arranged to receive a first initial state;

a primary-mask that is combined with an initial state of the auxiliary linear feedback shift register to produce a new initial state;

a first main linear feedback shift register that is arranged to receive the new initial state;

a second main linear feedback shift register that is arranged to receive a second initial state;

a set of fixed masks that are arranged to receive outputs from the first main linear feedback shift register and the second main linear feedback shift register;

a set of logic gates that are arranged to produce a set of outputs in response to the outputs from the first main linear feedback shift register and the second main linear feedback shift register and other outputs from at least one of the set of fixed masks, wherein the set of outputs are combined to produce a downlink scrambling code.

15. The downlink scrambling code generator of claim 14, further comprising a timing control that is arranged to trigger the auxiliary linear feedback shift register to pre-run when the downlink scrambling code to be produced corresponds to a secondary downlink scrambling code.

16. The downlink scrambling code generator of claim 15, wherein auxiliary linear feedback shift register pre-runs a number of samples that corresponds to a secondary index that is related to the downlink scrambling code to be produced.

17. The downlink scrambling code generator of claim 14, wherein the primary-mask corresponds to one of a set of primary masks that each correspond to a set of possible primary downlink scrambling codes.

18. The downlink scrambling code generator of claim 14, wherein the set of fixed masks allows memory allocated to storing the state of each mask in the set of fixed masks to be minimized.

19. The downlink scrambling code generator of claim 14, wherein the set of logic gates are at least one of AND logic gates and XOR logic gates.

20. The downlink scrambling code generator of claim 14, wherein the first initial state and the second initial state are set to correspond to any selected slot jumps.